Introduction to Programming for Scientists

Lecture 12
Parsing

Prof. Steven Ludtke
N410, sludtke@bcm.edu
Homework Review
Parsers

- Compilers/Interpreters
- Mathematical expressions
- Natural language
I run fast.
I’m going to go for a run.
The run queue on the computer is full.
Computer Languages

Reserved Word Lists (keywords)

- **Python**
  - and, as, assert, break, class, continue, def, del, elif, else, except, exec, finally, for, from, global, if, import, in, is, lambda, not, or, pass, print, raise, return, try, while, with, yield

- **C++**
  - and, and_eq, asm, auto, bitand, bitor, bool, break, case, catch, char, class, compl, const, const_cast, continue, default, delete, do, double, dynamic_cast, else, enum, explicit, extern, false, float, for, friend, goto, if, inline, int, long, mutable, namespace, new, not, not_eq, operator, or, or_eq, private, protected, public, register, reinterpret_cast, return, short, signed, sizeof, static, static_cast, struct, switch, template, this, throw, true, try, typedef, typeid, typename, union, unsigned, using, virtual, void, volatile, wchar_t, while, xor, xor_eq
Parsing Math

\[2 \times 3 \times 25 + 4^3\]

\[+\]

\[\times\]

\[2 \times 4 \times 3\]

\[3 \times 25\]
Parsing Math

2*(3*25+4^3)

How do we generate this?
Regular expressions?

http://re-try.appspot.com/
Parsers

- Lexical analysis
- Search for tokens
- Parsing or Syntactic Analysis
- Relate tokens to a ‘formal grammar’
- Evaluate Parse Tree
- Recursion !
Parsing


C/C++

Bison

Python

http://wiki.python.org/moin/LanguageParsing

We will use PLY (http://www.dabeaz.com/ply)